Formalization of Concurrent Performance Requirements in Building Problem Composition

Ardeshir Mahdavi, Ömer Akin, and Ye Zhang
Carnegie Mellon University
School of Architecture
Pittsburgh, PA
U.S.A.

ABSTRACT

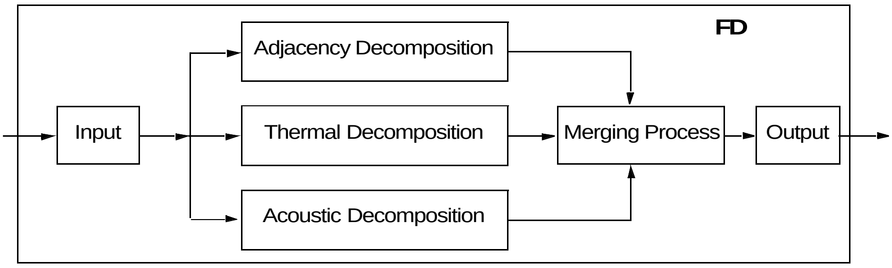
Specification of performance requirements is an emerging area of research that promises to improve building design particularly during the early stages of design. Building problem decomposition and recomposition can be based on a number of requirement categories in order to group building functions into hierarchically organized groups. Traditionally this activity is known as stacking and blocking, or zoning; and limited to spatial requirements. Our long term objective is to broaden this set into a more comprehensive one, including thermal, acoustic, and daylighting; and improve the state-of-the-art in building performance specification. While domain information from various building performance areas may be applicable toward enriching the informational basis for stacking and blocking operations, this paper focuses primarily on the thermal and acoustic domain.

1 INTRODUCTION

This paper introduces a generative design system to support functional stacking and blocking solutions based on multiple design requirements, namely functional adjacencies, as well as thermal and acoustic requirements. In order to formulate a computational model to process these design requirements and automatically generate stacking and blocking designs, two efforts were necessary. The first effort was aimed to identify, in addition to adjacency requirements, useful thermal and acoustic parameters for the process of stacking and blocking. This was done through protocol analysis of design processes (Akin 1978), parametric energy simulation studies, and acoustic problem analysis. The second step was to represent the building design requirements in computational form so that they could be automated.

Thus, the research toward the development of a computational stacking and blocking program, called functional decomposition in architecture (FD), involves three decomposition components to handle adjacency, thermal, and acoustic requirements. These are adjacency decomposition, thermal decomposition, and acoustic decomposition respectively. The FD system architecture containing the three decomposition components

Figure 1: **FD architecture**



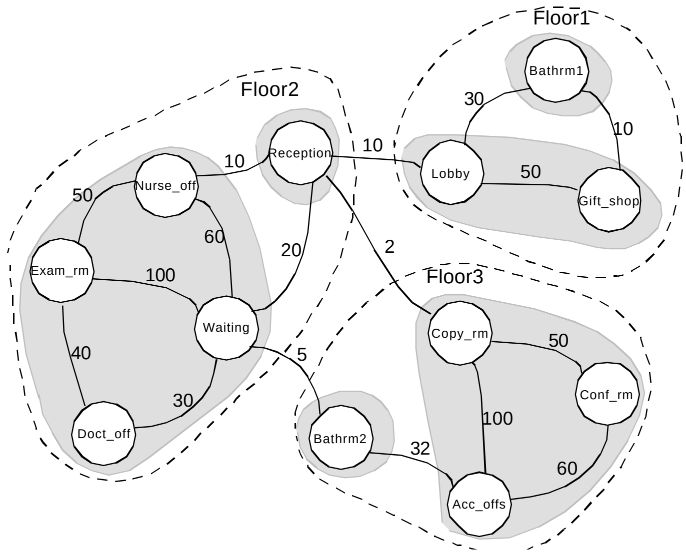
is shown in Figure 1. The objective of the "merging process" is to combine outputs from the three decomposition components. While it is attractive, from a theoretical point of view, to automate the merging process, no computational engine exists at this time for such purpose. Furthermore, the theoretical basis for combining multiple criteria does not exist. The tradeoffs between criteria, say acoustics and adjacency, are neither well understood nor consistent over different design problems or practices. Thus, the resolution of concurrent multi-domain decompositions is to be performed by the designer. Alternatively, the decomposition may be performed in one step based on a unified weighted set of pertinent adjacency, thermal, and acoustic criteria. We will discuss this approach in future publications.

In FD, there are two important terms: functional unit and functional unit hierarchy. These concepts were introduced in the development of a comprehensive design system intended for the early stages of building programming and design: SEED - Software Environment to support the Early phases in building Design (Akin et al. 1995). A functional unit is a construct corresponding to the spatial constituents of a building capturing a variety of design requirements applicable to it. A functional unit can be a room, a zone, or a floor. For the rest of the paper, a functional unit will be abbreviated as FU. A FU hierarchy is a hierarchical structure representing how a building is spatially organized. Each component in a FU hierarchy is a FU. A FU spatially contains all its constituent FUs.

2 ADJACENCY REQUIREMENTS

The goal of adjacency decomposition is to produce groupings of FUs for floors or to generate zones according to both area constraints and functional adjacencies in order to facilitate the designers' task of improving inter-space relationships.

Figure 2: **FUs with their relations (example)**



There are two types of adjacency requirements: strict adjacency and distance relations. Strict adjacency refers to geometric relations requiring FUs to be located right next to each other. Distance relations refer to geometric relations requiring FUs to be located within a certain distance. In FD, adjacency requirements are represented as a relation between a pair of FUs.

Thus the input to adjacency decomposition is a collection of FUs with size constraints for each FU and adjacency constraints between any pair of FUs. The example in Figure 2 illustrates the adjacency constraints in adjacency decomposition. The nodes represent FUs. The edges represent adjacency relations between two connected FUs. The numbers assigned to the edges show the strengths or weights of the adjacency relations - The larger the weights, the stronger the relations. The dotted areas represent floors. The FUs within each floor represent FUs grouped on that floor. The shaded areas represent horizontal zones. The FUs within each shaded area represent FUs grouped into that zone.

The FUs with strong relations should be grouped into the same floor and/or zone, while the FUs with weak relations could be separated into different floors and/or zones. More specifically, the goal of adjacency decomposition is to minimize the total weight between different floors/ zones, thus increasing the cohesiveness of FUs within a same floor/zone.

The process of adjacency decomposition includes three separate steps: stacking, horizontal zoning, and vertical zoning. The need for vertical zoning arises from the existence of vertically aligned FUs, such as bathrooms and staircases. Vertically aligned FUs form vertical zones.

Stacking

In the stacking step, adjacency decomposition groups FUs by floor in order to minimize the total weight between different floors. An objective function for stacking is as follows:

$$Objective = \min\left(\sum_{i=1, j=2, i<j}^{i=Nf-1, j=Nf} Weight(Floor(i), Floor(j))\right)$$

In the above formula, Nf is the number of floors in a building.

In the example of Figure 2, the total weight between the floors is 10+2+5=17. No other way of grouping the FUs can achieve a smaller total weight between the floors, subject to meeting the pre-defined area constraints of each floor. For this particular example, it is assumed that the number of floors and area assigned to each floor are predetermined.

Horizontal zoning

In horizontal zoning, adjacency decomposition groups FUs on the same floor by zone. FUs connected by stronger relations or larger weights will be grouped into the same zone, subject to certain zoning constraints such as pre-defined number of zones on that floor.

In the example in Figure 2, assuming the user wants two zones on Floor1, Lobby and Gift_shop are assigned to the same zone whereas Bathrm1 is a separate zone by itself, because the weight between Lobby and Gift_shop (50) is greater and thus they should be grouped.

Besides the required number of horizontal zones, there are two other optional methods of deciding at what points should FUs be separated into different horizontal zones. The following is a discussion of these three methods :

- The user knows how many zones are needed and the relative area of each zone. But this is unlikely in the early stage of design.
- The adjacency relations between the FUs suggest a partition pattern themselves. In this case, an algorithm will be necessary to find such a pattern.
- There is some threshold value that the user wants to define. This applies to the situation when the user knows what adjacencies are relatively more important than others and is able to express this in the form of a threshold value.

Adjacency decomposition will provide these three options for the user.

Vertical zoning

For vertical zoning, FUs with vertical relations are represented as vertical zones. Such vertical relations include vertical plumbing connections (e.g., bathrooms), vertical traffic connections (e.g., staircases), and vertical space connections (e.g., atria).

Output

The output of adjacency decomposition represents which FUs should be grouped into the same floor/zone, and which FUs should be separated. Figure 3 shows the output structure of stacking and horizontal zoning for the example shown in Figure 2. The shaded FUs represent FUs given as original input, i.e., nodes in Figure 2. The links represent spatial containment relations; e.g., Floor1 spatially contains Bathrm1 and Zone1. The input FUs are aggregated into floors and again into zones and possibly into sub-zones. The same hierarchical structure can be applied to the outputs of all the other decomposition components as well.

Figure 3: **Output of adjacency decomposition (example)**

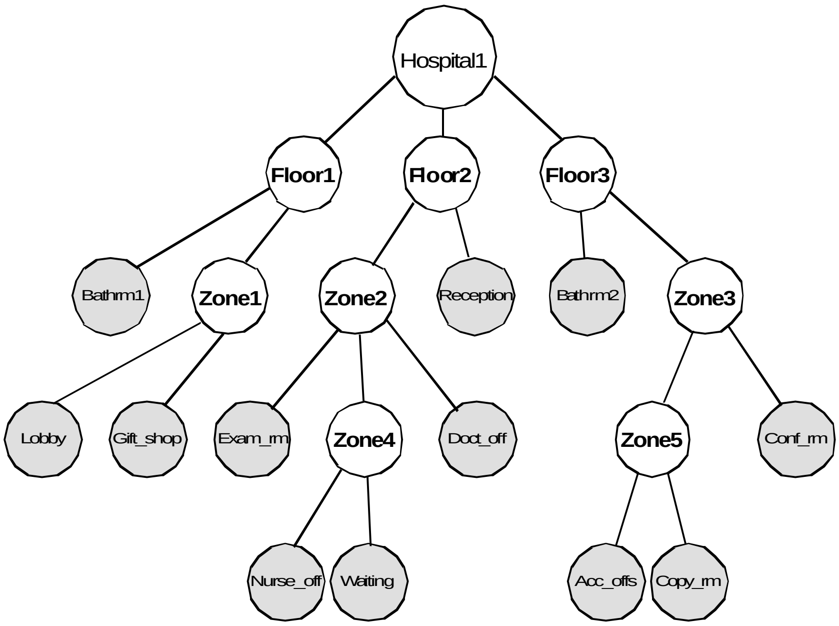
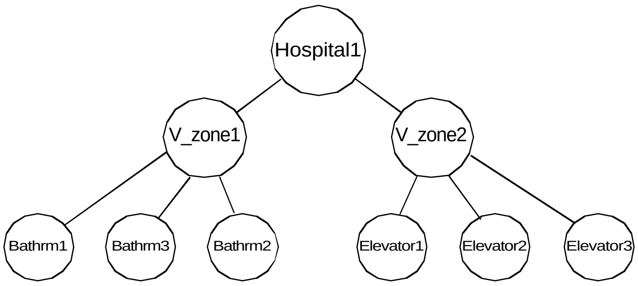


Figure 4: **Output of vertical zoning (example)**



Additionally, the output of vertical zoning represents vertical zones in a building. Figure 4 shows such an output structure for the example in Figure 2. For instance, the bathrooms on each floor form one vertical zone (V-zone1) and will thus be vertically aligned during the stage of layout generation (which is not covered in this paper).

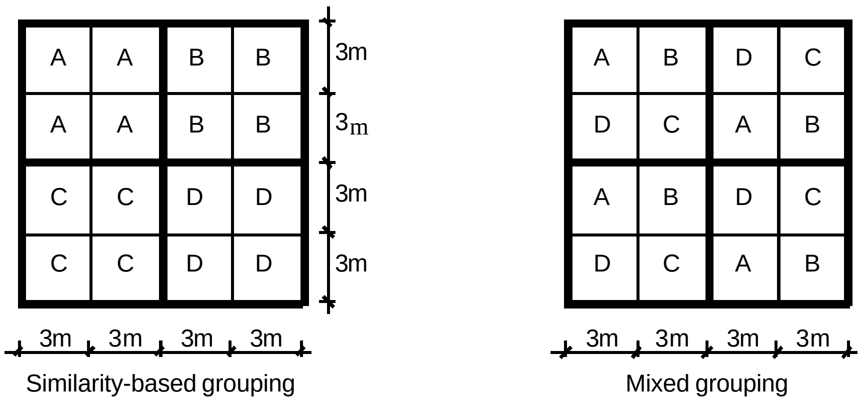
3 THERMAL REQUIREMENTS

Parametric energy simulation studies were conducted using the building performance modeling software SEMPER (Mahdavi 1996) in order to identify useful thermal parameters for functional grouping. Four parameters were tested: space temperature, load schedule, internal load, and air exchange rate.

Toward this end, for each of the four parameters, a single floor building with a similarity-based grouping condition was tested against a single floor building with a mixed grouping in order to compare their annual energy use (Figure 5). The idea was to examine, if grouping of FUs with similar thermal characteristics (e.g., similar desirable air temperature or similar minimum required air exchange rates) will result in significant differences in the predicted resulting performance, i.e., in this case, the predicted annual energy use. In principle, thermal requirements include energy consumption, environmental impact, and thermal comfort. This work focused on energy consumption only.

Two principles were followed in this simulation study. The first principle was to separately test each thermal parameter, i.e., when testing a parameter, the remaining thermal parameters were kept constant so that they would not affect the results for the tested parameter. The second principle was to make all the other factors irrelevant. This effect was achieved by making each FU the same size and uniform square shape, designing each group so that each had the same number of FUs, and making locations/orientations irrelevant by not considering solar access.

Figure 5: **FU arrangements considered for the thermal simulations (A, B, C, and D are different values of a thermal parameter)**



Obviously the perimeter versus core location of a FU has a significant impact on daylighting penetration and solar gain. While these issues were not considered in the present study, work is underway to make FD responsive to the FUs' relationship to external environment. We plan to do this via inclusion of the external environment as a separate FU which can possess different coupling levels to the building program's FUs.

The analysis of the results of this specific set of simulations, i.e., eight annual energy consumption levels, using the ANOVA technique, showed that only grouping based on the temperature values significantly affects the predicted energy use. Temperature may be thus regarded as a useful thermal parameter for grouping FUs. Therefore temperature was selected as an example to illustrate a formalization of the thermal decomposition process.

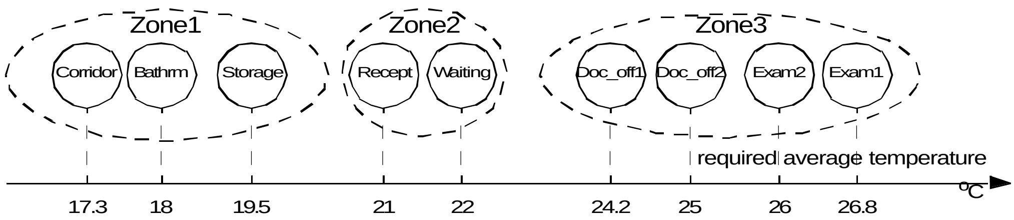
The goal of thermal decomposition is to group FUs by zone according to thermal requirements in order to maximize energy efficiency. In mathematical terms:

$$Objective = \min\left(\sum_{i=1}^{Nf} \sum_{j=1}^{Nz(i)} energyUse(Zone(i, j))\right)$$

In the above formula, Nf is the number of floors in a building; Nz(i) is the number of zones on floor i; Zone(i, j) is zone j on floor i.

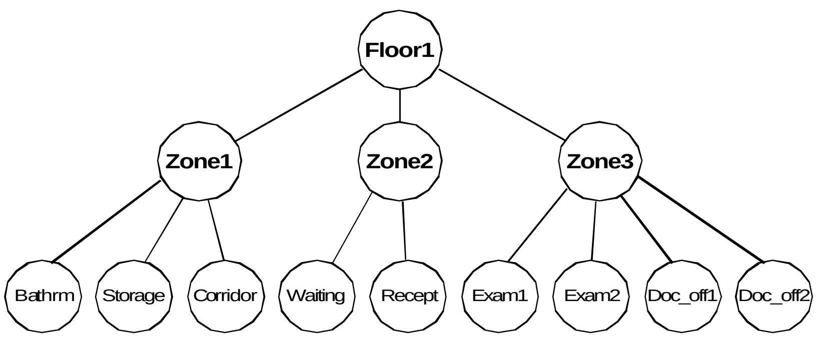
The input to temperature decomposition is a collection of FUs with each FU having area and minimum/maximum temperature requirements.

Figure 6: **Thermal zones consisting of FUs with close average temperature requirements**



Once the proximity of the space temperatures is translated into coupling weights between FUs, stacking and blocking operations may be performed analogous to adjacency-based decomposition. However, in thermal decomposition, a computationally less expensive approach may be selected, since the grouping does not need the explicit definition of coupling weights between FUs. Thus, FUs on each floor can be grouped into zones according to the average (mean value) temperature requirements. FUs with similar average temperature requirements are grouped into the same zone, and vice versa. Figure 6 shows an example of how to group FUs into zones. In Figure 6, the nine nodes represent nine FUs already assigned to the same floor. To group the FUs into zones, the FUs are positioned in the one-dimensional Euclidean space according to their required average temperatures. The distance between any pair of FUs shows the difference in their required average temperatures: the closer they are, the more likely they are to be grouped into the same zone. It is thus concluded that temperature decomposition is a problem of grouping FUs according to their individual characteristics, i.e., average temperatures.

Figure 7: **Output of thermal decomposition (example)**



The output of thermal decomposition is, in this case, a recommendation of which FUs should go to the same thermal zone on a floor and which FUs should be separated. The output structure of thermal decomposition, like that of adjacency decomposition, is a hierarchical structure. Figure 7 shows the output of the example shown in Figure 6.

4 ACOUSTIC REQUIREMENTS

The goal of acoustic decomposition is to produce groupings of FUs to zones in order to satisfy the FUs' acoustic requirements at minimum construction cost. The corresponding mathematical objective function is as follows:

$$Objective = \min(\sum_{k=1}^{Nz} \sum_{i<j}^{FU(i),FU(j)\subset Zone(k)} constructionCost(FU(i), FU(j)))$$

In the above formula, Nz is the total number of zones on a floor.

The key issue is to minimize construction cost. Theoretically FUs with any noise levels can be adjacent to each other as long as appropriate noise control technologies are used to reduce the acoustic interference between adjacent FUs. The goal of minimizing construction cost is realized through the arrangement of FUs within a building, either by grouping acoustically compatible ones or by separating ones that interfere with each other.

For the purpose of this research, three acoustic parameters are important: *a)* STC or sound transmission class describing the degree of decoupling between two adjacent FUs for air-borne sound, *b)* IIC or impact isolation class describing the degree of decoupling between two vertically adjacent FUs for structure-borne sound, and *c)* NC or noise criteria defining the permissible noise level in a FU.

In order to solve acoustic decomposition, the above parameters are considered in terms of two criteria: *a)* EEL or expected emission level, and *b)* PSL or permissible sound level. To ensure that the maximum PSL requirements of a FU is met, the required air-borne and structure-borne decoupling level ($STC_{required}$, $IIC_{required}$) between this FU and an adjacent FU can be estimated as follows:

$$STC_{required} = EEL_{airBorne} - PSL + 8$$

and

$$IIC_{required} = EEL_{structureBorne} - PSL + 8$$

Table 1: **Discrete categories and numeric examples for EEL and PSL**

Scale Value	EEL	dB	PSL	dB
3	EL	105	EI	75
2	VL	90	VI	65
1	L	75	I	55
0	N	60	N	45
-1	Q	45	S	35
-2	VQ	30	VS	25
-3	EQ	15	ES	15

Here $EEL_{airBorne}$ refers to the adjacent FU's output noise level, and $EEL_{structureBorne}$ refers to the impact noise level from the adjacent FU. In the above equations, the constant
value of 8 decibel is added to the $STC_{required}$ and $IIC_{required}$ in order to ensure that the PSL value in the receiver FU is not significantly affected by the energy transmitted from the source FU.

As for the format of each of the above mentioned attributes, acoustic decomposition provides, in this case, two forms of user input, namely discrete categories and/or actual decibel values as shown in Table 1. This will provide flexibility for the user.

The discrete attributes for EEL, as shown in Table 1, include extremely loud (EL), very loud (VL), loud (L), neutral (N), quiet (Q), very quiet (VQ), and extremely quiet (EQ). The discrete attributes for PSL, also shown in Table 1, include extremely insensitive (EI), very insensitive (VI), insensitive (I), neutral (N), sensitive (S), very sensitive (VS), and extremely sensitive (ES). The decibel values represent illustrative sound levels of the corresponding discrete values which must be determined on a case by case basis.

Table 2 contains four examples of discrete acoustic requirements by FUs to concretely illustrate the level of loudness/sensitivity of the discrete EEL and PSL values.

Table 2: **Four FUs' discrete acoustic values (example)**

	EEL (air-borne)	EEL (structure-borne)	PSL
Kitchen	VL	L	I
Symphony Hall	VL	VL	ES
Machine Room	EL	EL	EI
Reading Room	Q	N	VS

In this paper, acoustic decomposition includes horizontal decomposition involving dividing the collection of FUs into zones with minimum construction cost. Vertical decomposition is formulated into the task of specifying required construction costs when a FU is located directly on top of another. In this way, appropriate constraints can be set to guide layout generation toward favorable design solutions.

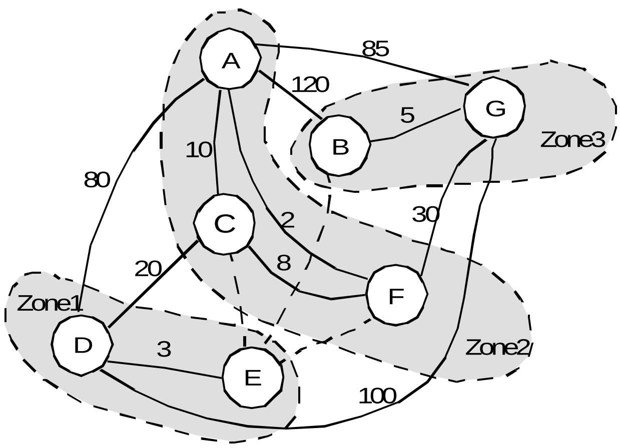
Horizontal decomposition

In acoustic decomposition, horizontal decomposition is realized in two steps, namely required construction cost calculation and formation of a graph, and partition of the graph. In step one, in order to calculate required construction cost between any pair of FUs, $STC_{required}$ needs to be calculated as an intermittent step. Each pair of FUs' $STC_{required}$s will be calculated according to their $EEL_{airBorne}$ and PSL. Among the two calculated $STC_{required}$s, the greater one is the required decoupling between the two FUs, therefore it will be used as a basis in calculating the required construction cost between the two FUs. When the construction costs between all pairs of FUs are calculated, a graph of FUs with edges of different construction costs will be formulated similar to the one shown in Figure 8.

It is assumed here, for demonstrative purposes, that a required construction cost is dependent on two factors, namely $STC_{required}$ and size of the shared wall between the two FUs. When the height of walls is considered constant, L_{wall}, the length of the shared wall between the two FUs, can be used to estimate the size of the shared wall:

$$L_{wall} = \min(sqrt(Area_{FU1}), sqrt(Area_{FU2}))$$

Figure 8: **Graph containing FUs and partition of FUs into acoustic zones (Dotted lines show unknown relations due to lack of input)**



Obviously, construction cost is a complex function of STC. A highly simplified assumption would be to correlate cost with material use (expressed in terms of surface density *m* in kg.m^{-2}) and use an approximate function for the relationship between STC and m:

$$STC \cong 32.4log(m) - 26$$

Assuming χ is a cost coefficient to be determined on a case by case basis, the following relation can be derived:

$$ConstructionCost = c \times 10^{(\mathrm{STCrequired}+26)/32.4} \times L_{wall}$$

This equation is used to derive required construction cost due to the partition element between a pair of adjacent FUs.

In step two, a graph-partitioning algorithm will be run on this acoustic graph to generate zones. Those requiring greater construction costs for decoupling should be separated into different zones in order to minimize construction cost; those requiring smaller construction costs for decoupling should be grouped into the same zone, since they do not need much decoupling when allocated together.

<u>Vertical decomposition</u>

Vertical decomposition is to calculate and represent required construction costs between any pair of FUs when they are located on top of other. The processes of vertical decomposition include calculating $IIC_{required}$ and $STC_{required}$ between any pair of FUs if one is to be located on top of the other, and calculating the corresponding required construction costs and representing them using a graph.

To calculate $IIC_{required}$ and STC $_{required}$, assuming FU-A is located on top of FU-B, the following relations can be derived similar to the equations in horizontal decomposition:

$$STC_{required(B \rightarrow A)} = EEL_{B,\,airBorne} - PSL_{A} + 8\,,$$

$$STC_{required(A \rightarrow B)} = EEL_{A,\,airBorne} - PSL_{B} + 8\,,$$

and

$$IIC_{required} = EEL_{A,\,structureBorne} - PSL_{B} + 8$$

Among the two calculated STCs ($STC_{required(B⌘A)}$ and $STC_{required(A⌘B)}$), the greater one is the required decoupling between the two FUs.

It is assumed here that a required construction cost is related to two factors, i.e., construction cost resulted from $IIC_{required}$ and construction cost resulted from STC $_{required}$. We have already discussed an example for a simplified approach to derive construction cost estimates based on the required STC levels. Again, if we substantially simplify matters, we may correlate cost with material and use an approximate function to establish a relationship between IIC and surface density:

$$IIC \cong 35 log(m) - 54$$

The construction cost resulting from $IIC_{required}$ could be then calculated as follows:

$$ConstructionCost = g \times 10^{(IICrequired+54)/35} \times Area_{shared}$$

In this equation, $Area_{shared}$ is the shared floor/ceiling area between the two FUs, i.e., the smaller area of the two FUs', and γ is a coefficient that must be determined on a case by case basis.

The construction cost resulting from $STC_{required}$, similar to the horizontal decomposition case, is calculated as follows:

$$ConstructionCost = c \times 10^{(STCrequired+26)/32.4} \times Area_{shared}$$

To unify the construction cost implications of the air-borne and structure-borne decoupling requirements, for the vertical decomposition, the following procedure may be followed. First, the minimum necessary surface density is calculated to fulfill both STC and IIC requirements for a FU-dividing partition element. Second, the larger of these two surface density values is selected to modify either the STC or IIC requirements. Third, these modified requirements are used to derive the pair of construction costs. Fourth, the higher of the two construction costs is used in a graph representation.

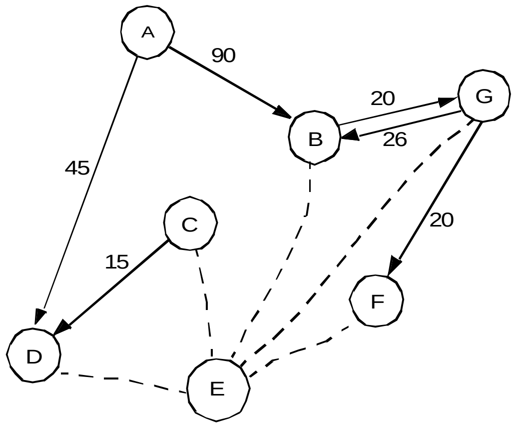
Output

The output of horizontal decomposition is represented in the form of a FU hierarchy.

A typical output for vertical decomposition is similar to the one shown in Figure 9. It is a graph with weights on edges showing required construction costs in relative units. Specifically, a weight between two adjacent FUs shows the required construction cost for decoupling if the "from" FU is located right on top of the "to" FU. For example, in case of FU-A which is located right on top of FU-B, 90 cost units will be needed.

In the case of using FD's findings in layout generation, this output would take the form of a set of constraints to be observed when generating layouts for two vertically

Figure 9: **Cost graph of vertical placement of FUs (Dotted lines show unknown relations due to lack of input)**



adjacent floors. Here, each edge shown in Figure 9 would be expressed as a constraint (relationship) between two FUs.

5 ALGORITHM DESIGN

5.1 **Stacking Algorithm in Adjacency Decomposition**

In the stacking process in adjacency decomposition, the goal is to minimize the total weight between different floors. The cumulative magnitude of weights between different floors is called cut size. An objective function for stacking is as follows:

$$Objective = \min\left(\sum_{i=1, j=2, i<j}^{i=Nf-1, j=Nf} Weight(Floor(i), Floor(j)) \right)$$

In the above formula, Nf is the number of floors in a building.

It is assumed that the user has specified required number of floors Nf and each floor's area. All the FUs are sorted by their total connected weights in descending order. Each of the first Nf-1 FUs in this sorted list is assigned to a separate floor. All the remaining FUs are assigned to the last floor. This becomes an initial state with unbalanced floor areas. Then some FUs on the last floor will be moved to the other floors according to the weights between them and FUs assigned to the other floors. Once a FU is moved, it will be locked and will not be moved again. As FUs are moved gradually from the last floor to the other floors, the balance of the areas between the floors is improved.

The criteria in selecting a move is the expected decrease in the cut size (gain) between different floors. The move which will result in a maximum gain will be chosen. While comparing all the gains of unlocked FUs in order to select a move is time-consuming, a data structure "bucket array" is used to record gains of all the unlocked FUs for efficient retrieval of a best move. This bucket array is an extension to a bucket list that Fiduccia and Mattheyses (1982) used in their graph-partitioning algorithm.

Each time a FU is moved, a new state is generated which is a partition of the FUs to the floors. When either all the FUs are locked or no move will improve the area balance between the floors, the process is over. Among all the states that are generated, the one which satisfies the required floor areas and has the minimum cut size will be chosen as a solution to the stacking problem.

5.2 **Deciding Optimal Stacking Order Algorithm in Adjacency Decomposition**

The final result of the stacking algorithm in adjacency decomposition is a set of floors each of which satisfies a required floor area. When all the floors' required areas are different, the stacking order is obvious. For instance, if a building has three floors and floor one, floor two and floor three's areas are 2000, 2200, and 1500 respectively, after running the algorithm, there will be three generated floors A, B, and C with areas 1500, 2000, and 2200 respectively. It is obvious that B is floor one, C is floor two, and A is floor three. But when some or all the floors are of the same area, a procedure will be necessary to determine a stacking order among the generated floors, subject to meeting the following objective function:

$$Objective = \max\left(\sum_{i=1}^{Nf-1} Weight(Floor(i), Floor(i+1))\right)$$

In the above formula, Nf is the number of floors in a building.

This is a problem of identifying an optimal mapping between a set of generated floors (candidate floors) and a set of floors with required areas (target floors) that has the maximum total weight between all the adjacent floors. A dynamic programming algorithm is used. The stacking order is decided from floor one up to the top floor. In order to decide an optimal mapping for the floors, all the mapping alternatives for each target floor is considered. When all the mapping alternatives for all the target floors are considered, the set of mapping with the maximum total weight between adjacent floors is selected as an optimal mapping. Stacking order can be decided correspondingly.

5.3 **Zoning Algorithm in Adjacency Decomposition**

In adjacency decomposition, FD zones FUs on each floor based on the so-called "Union Find" method. The general process of Union Find is, in a graph-partitioning problem, to group nodes into clusters according to certain criteria, and gradually merge clusters until some termination condition is satisfied. The criteria chosen in zoning is to group FUs with largest weights. This process guarantees that FUs related with large weights are grouped subject to meeting some specified constraints.

In this zoning algorithm, initially each FU is assigned to a separate zone. All the edges are sorted in descending order according to their weights. Then all the edges are visited in the sorted order. In the process of traversal of all the edges, if two linked FUs are in different zones, their zones will be merged if doing so does not violate the required zone number and size. This process terminates when any of the following conditions holds: *a)* the user-specified constraint (such as required zone number and size) is satisfied, *b)* an edge is met with a weight larger than a threshold weight that the user has defined, and *c)* all the edges have been visited.

5.4 **Zoning Algorithm in Thermal Decomposition**

The objective function which is used to evaluate a zoning state or to compare different candidate moves is as follows:

$$Objective = \min\left(\sum_{i=0}^{Nz}\left(\left(\sum_{j=0}^{Nfu(i)}\left(temperature(j) - center(i)\right)\right) / \, numFUsInZone(i)\right)\right)$$

In the above formula, Nz is the number of zones on the floor; Nfu(i) is the number of FUs in zone i.

It is assumed that the user has specified required number of zones and maximum zone size. The set of FUs on a floor are broken into the specified number of zones subject to meeting the specified area constraints in a fashion that FUs with contiguous average temperature values are assigned to the same zone. Then the zones are adjusted by moving a border FU at a time from its current zone to its adjacent zone. The way a FU is selected is by comparing the improvements over the objective value when each of all the border FUs is moved to its adjacent zone. The move that will minimize the objective value will be chosen. This process terminates when there are no moves remain which will improve the objective value.

5.5 **Zoning Algorithm in Acoustic Decomposition**

The goal of acoustic decomposition is to minimize the sum of the total weight within each zone. A corresponding objective function is as follows:

$$Objective = \min\left(\sum_{k=1}^{Nz} \sum_{i<j}^{FU(i),FU(j)\subset Zone(k)} Weight(FU(i), FU(j))\right)$$

In the above formula, Nz is the number of zones on a floor.

For acoustic decomposition, a similar procedure of graph-partitioning as the one used in adjacency decomposition may be applied.

6 CONCLUSION

This work presented a method for handling thermal and acoustic requirements besides adjacency requirements in the stacking and blocking design process. First, for each domain illustrative objective functions were established. Second, for each domain the relevant performance requirements were identified. Third, a process was established to derive numeric weighting values for the use of partitioning algorithms toward spatial stacking and zoning. Future work will include: *a)* the extension of performance criteria to daylighting requirements, *b)* a unified graph solution for an aggregate FU scheme based on multiple performance criteria, and *c)* empirical usability tests of the FD engine.

7 REFERENCES

Akin, Ö. (1978) How Do Architects Design? in Latombe (ed.), *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, North-Holland Publishing Company, pp. 65-98.

Akin, Ö., R. Sen, M. Donia, and Y. Zhang (1995) SEED-Pro: Computer-Assisted Architectural Programming in SEED, *Journal of Architectural Engineering*, ASCE, December, 1995, vol. 1, no. 4, pp. 153-161.

Fiduccia, C. M. and R. M. Mattheyses (1982) A linear Heuristic for Improving Network Partitions, *Proc. ACM/IEEE Design Automation Conf.*, pp. 175-181.

Mahdavi, A. (1996) SEMPER: A New Computational Environment for Simulation-based Building Design Assistance, *Proceedings of the 1996 International Symposium of CIB W67 (Energy and Mass Flows in the Life Cycle of Buildings)*, Vienna, Austria, pp. 467-472.